H2020-PHC-2014-single-stage
PHC-26-2014: Self management of health and disease: citizen engagement and mHealth
Research and Innovation action

Mobile Therapeutic Attention for Patients with Treatment Resistant Schizophrenia

Deliverable 3.8

m-RESIST Prototype V0

Version	1.0.0
Status	Final
Date	09/06/2017
Filename	643552_m-RESIST_20170609_D3.8_m-Resist_Prototype V1.0.0
Type	Demonstrator
Dissemination level	Public

Project reference No.	643552
Work package	WP3: Technical Design and Development

Authors

Date	Name	Partner
Document leader	Fotis Paraskevopoulos	ICCS
Participants		
	Jesús Berdún/Elisenda Reixach	TICSALUT
	Yoram Feldman	GERTNER
	David Vermeir	imec
	Xavier Constant	IBM
	Anna Triantafillou, Panagiotis Kokkinakis	ATC
	Valentina Simonetti	AB.ACUS
	Maria Bulgheroni	AB.ACUS
	Brunella Marzolini	AB.ACUS
	Iluminada Corripio	IR-HSCSP
	Eva Grasa	IR-HSCSP

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 643552"

Document History

Version	Author	Change	Status
0.0.1	Fotis Paraskevopoulos	Outline and first proposal on procedures	Draft
0.0.2	Fotis Paraskevopoulos	Contributions	Draft
0.0.3	Valentina Simoneti	Contributions	Draft
0.0.4	David Vermeir	Contributions	Draft
0.0.5	Fotis Paraskevopoulos	Contribution Harmonization	Draft
0.0.6	Panagiotis Kokkinakis	Contributions	Draft
0.1.0	Jesús Berdún	General contributions and Review	Draft
0.1.1	Fotis Paraskevopoulos	Review and Conclusions	Draft
1.0.0	Jesús Berdún/Elisenda Reixach	Final review and harmonisation	Final

Table of Contents

Table of figures

1 Introduction

1.1 Purpose of the document

The following document presents deliverable D3.8 “m-RESIST Prototype V0”. m-RESIST is a mobile ICT solution addressed to empower patients with treatment-resistant schizophrenia (TRS), that encourages the patients and caregivers to actively participate in the therapeutic process and enables them to self-manage their condition. This deliverable is classified as a “demonstrator”, and shows an improved version of the first m-Resist Beta Prototype after testing activities performed during a pre-trial phase with healthy users in WP4. This version will be used for the piloting activities with real patients in WP5 in the three piloting countries (Spain, Hungary and Israel). The following document presents an overview of the system, describing the high level architecture and the different functionalities that it offers.

This deliverable does not provide an architectural description of each component, as this description has already been provided in previous WP3 deliverables. Rather, it presents the work done in terms of development, and the state of the current prototype, briefly describing each component of the system, as well as the functionality which is ready and integrated.

1.2 Relation with other deliverables

D3.8 is the “demonstrator” deliverable of the m-RESIST Prototype V0, the resulting system from the development carried out after the testing of m-Resist Beta Prototype performed during WP4 activities. The m-RESIST Prototype V0 will be the system version that will be used for piloting activities with real patients in WP5. These results will be used to improve the system and deliver the D3.9 m-RESIST Prototype V1 that will be delivered at the end of the project. In this document we have included excerpts from previous technical deliverables (D3.1-7) providing a more detail overview of the architecture and technical components of the system.

2 m-RESIST Overview

2.1 m-RESIST tools

The m-RESIST solution is an innovative disease management system based in mobile technologies, addressed to empower patients with treatment-resistant schizophrenia, which will engage them, together with the professionals and caregivers, in active participation in therapeutic processes, enabling them to better self-manage their condition.

Figure 1: m-RESIST system tools

m-RESIST provides a different set of tools and functionalities addressed to patients, caregivers and health professionals, which rely on three main tools: two user interfaces and one back-end system. These tools are: first, a mobile app installed in **a smartphone connected to a smartwatch** for patients and caregivers; second, a **web-based dashboard** for follow-up and monitoring for clinicians; and third, a **back-end system for managing patient data**, interventions and communication between patients, caregivers and clinicians (m-RESIST back-end system).

The functionality of the **smartphone** is based on sensor data collected (communication and sleep patterns, the conference tool to perform online visits with clinicians, and the m-RESIST mobile application. This app has the goal to support patients in managing their health and caregivers in improving their knowledge about the disease and their communication with clinical staff. This is a multifunctional mHealth app address to: i) inform, by providing psychoeducational information related with the illness, and reports keying off their progress based on their own data in a variety of formats (text, photo, video) ; ii) instruct, by providing instructions to the user; iii) record, by capturing user entered data and sensors data; iv) display: graphically display/output user entered data; v) guide, by providing guidance and recommendations based on user entered information; vi) remind/alert: by providing reminders to the user; vii) communicate, by providing communication between healthcare providers and patients/caregivers.

The wearable used in m-RESIST consists in a **smartwatch** that collects data from the user to be wirelessly transmitted to the smartphone. Health data is recorded through automatic passive upload, mainly level of activity: movement, sleep pattern, steps counter and heart rate.

The **web-based platform** is the tool used by clinicians to monitor patients' state and review data collected by sensors, to communicate by messaging with patients, caregivers and other professionals, or to consult recommendations (based on guidelines and experts opinion). This platform is connected to the **back-end system** of m-RESIST, which consist in the following modules:

- Sensor module: collects sensor data from the smartphone and the wearable (smartwatch), processes it and stores it in the m-RESIST Information Repository, providing the needed information to trigger therapeutic interventions and to visualize and monitor the patient status through the dashboard.
- m-RESIST Information Repository: acts as the central data repository for the entire system, allowing other modules to store and retrieve any kind of information in a flexible format.
- Clinical Decision Support System (CDSS): rule engine that uses decision algorithms under given clinical conditions collected either automatically (by sensors) or manually (user interaction). Patient data is interpreted according to pre-determined rules, triggering series of actions in three therapeutic interventions, namely, symptom management, treatment adherence and healthy lifestyle.
- Recommender: using information stored in the system, it provides a set of recommendations addressed to the patient, caregiver or professional, according to therapeutics interventions. Recommendations also consider previous user feedback and outcomes from the predictive module.
- Predictive module: provides predictions for different measures of a patient's status based on the information currently available in the system about the patient. A set of models have been developed for the predicting variable "Patient Evolution".
- Integration layer: it conforms the internal structure of m-RESIST back-end that results in a system flexible enough to support the coupling and effective communication between various components.

2.2 m-RESIST Interventions

The m-RESIST system considers specific mHealth intervention modules specially designed for patients with treatment-resistant schizophrenia. The different interventions defined in the project have been divided in 2 categories, ***Basal modules*** (Treatment Adherence, Healthy Lifestyle, and Symptom Management-CBTp) that are oriented to develop abilities in the patient; and *Risk modules* (Symptom Management-Risk), that is oriented to deal with specific problems, especially in risk situations. Interventions are briefly described next:

Symptoms Management Intervention: it contains the basic set of questions that measure the patient's psychiatric profile on a frequent basis and triggers appropriate follow up questions depending on the patient's answers. This module can be functioning in two different ways, basal or risk. When the basal version is activated, intervention consists on learning to deal with symptoms and early warning signs by means of practicing coping strategies, according to protocol of cognitive behaviour therapy for psychosis (CBTp). When the risk version is activated, then the intervention is addressed to identify the risk behaviour (suicide behaviour, aggressive behaviour) that is present and help patient by

recommendations, messages or notifications, with early detection and immediate interventions if necessary. Significant sensors deviation will early detect relapses and trigger intervention.

Treatment Adherence Intervention: is targeted to measure and help to reinforce the patient's involvement in the treatment plan. When the patient is not adhering to the treatment, it tries to gather more information from the patient or caregiver. It makes use of reminders and psycho-educational content to help patients regain a connection with their treatment. Significant sensors deviation will early detect non-adherence and trigger interventions.

Healthy Lifestyle Intervention: The Healthy Lifestyle intervention is an intervention that assists the patient in resolving certain problems that are influenced by antipsychotic treatment or an unhealthy lifestyle. It could work as a supplementary tool for clinical intervention performed outside the system (for example, dietitian). This intervention is related with sensor data collected by the smartwatch. In case of significant changes in predefined thresholds of steps count and/or low activity, specific questionnaires, recommendations and notifications will be activated.

The basal modules will remain temporally deactivated when the system detects a risk situation and the activation of the risk module was necessary. Basal modules should function only when risk levels is < 2. Above this risk, the system should activate the risk module. The m-RESIST system can detect risk situations by itself (significant changes in sensor data thresholds, e.g. detecting low activity, sleep disturbances) or by patients or caregivers, if they use the "alarm bottom" to ask for help.

2.3 m-RESIST therapeutic process

The main aim of the m-RESIST therapeutic process is to engage TRS patients, together with the caregivers, in active participation in therapeutic processes, and at empower them, enabling to self-manage their condition. This process intends to be tailored to user needs, and the main actors involved in the deployment of the interventions are the psychiatrists, psychologists, case managers and nursing staff. The process to follow consists of 3 stages: training, pre-intervention and intervention.

1. Training: the participants are trained to ensure a good understanding of the m-RESIST tools, and to help them to feel comfortable with the devices (wearable and mobile application). They are trained by the mHealth support specialist on how to use the functions of the devices (i.e. using a touchscreen, call, text). Patients and caregivers are provided login credentials for the smartphone app, and research staff provides patients with information on how to use the app, including a help guide and other relevant documentation, as well as user support mechanisms.

2- Pre-intervention: the main aim is to reach an agreement between patient and the m-RESIST clinical team regarding the therapeutic details (list of aims to reach) and the scope of the monitoring (by using wearables), including caregivers in the process. During the baseline assessment, a personal relapse signature will be defined based on health-related data from the patient condition. This information will be interrelated with a list of coping strategies that will aim to help patients managing distress. At baseline assessment, each patient together with the caregiver and the psychotherapist will define this list of techniques. If the patient has own coping strategies, a "tailored" list to deal with EWS will be define. On the contrary, a list of general coping strategies will be determined. These recommendations can be a possible system response when patients express distress and need for help.

3- Intervention: The list of problems is created together by patient, caregiver and clinician. Depending on the problems selected, the treatment plan will be configured, and either **basal or risk modules** will be activated

2.4 Architecture

A diagram depicting the high-level functional architecture of the system is shown in **Error! No s'ha trobat l'origen de la referència.**. Real data flows are based on mIR updates and ESB communication mechanisms, which for simplification reasons have not been reflected in the diagram.

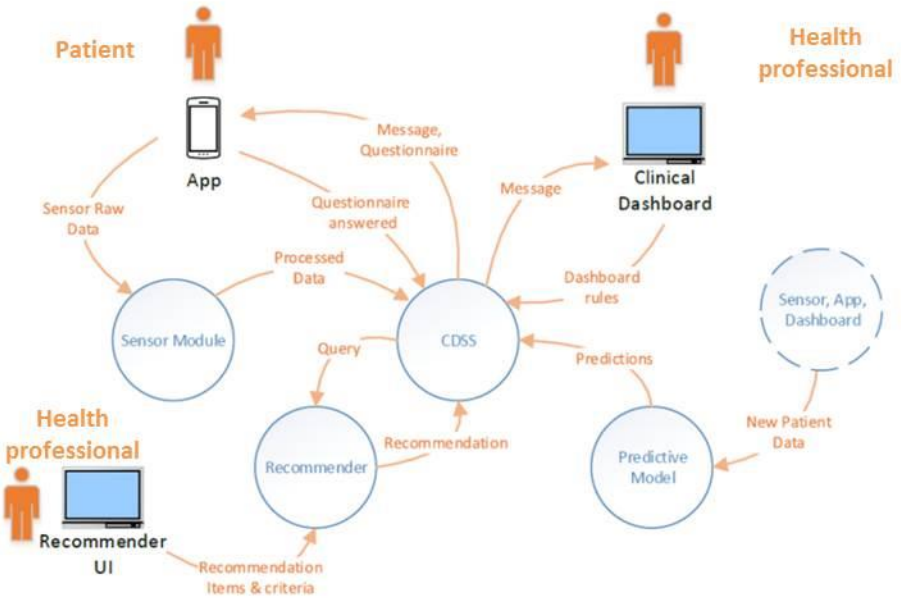

Figure 2: m-RESIST high-level functional architecture

The users (clinicians, patients and caregivers) interact with the system through the App, the Clinical Dashboard and the Recommender UI. The App, installed on the mobile devices, is the point of interaction between the patients and the system. The App, on one hand, delivers messages and questionnaires to the patient and sends the patients' answers back to the system. On the other hand, it gathers raw sensor data from the wearables and sends it to the system, where it is processed by the Sensor Module.

The CDSS (Clinical Decision Support System) is the module in charge of delivering messages and questionnaires to the App and retrieving their answers. These are generated through a rule-based system that monitors changes in the patients' baselines, using data from the Sensor Module, the collected questionnaires and the Predictive Module. The Predictive Module provides the CDSS with predictions and probabilities for the different possible outcomes of each patient based on its current and past status. If the CDSS decides a patient needs an intervention, it queries the Recommender for a

ranked list of possible interventions for that patient. These can be configured by the clinicians through the Recommender UI.

The clinicians can access patient data through the Clinical Dashboard. The Dashboard allows the clinicians to visualize patient statistics, predictions, recommendations and messages and to interact with patients and other clinicians. It also allows the clinicians to adjust the CDSS rules to meet the patients' needs.

2.5 Testing scenarios

In order to exemplify the use of the m-RESIST solution, in this section there is a description of two use-case scenarios that shows the functioning of the system, related to the piloting activities planned in WP5. A set of activities are planned during the "pre-intervention period" and the "intervention period", which are aimed to: first, recruit patients, train them, make a baseline assessment and define the therapeutic treatment; and second, to use the m-RESIST solution to perform a follow-up, monitoring and exit of the patients through the m-RESIST tools.

The first scenario described explains a pre-intervention period activity, namely, patient recruitment, assessment and therapeutic plan. The second scenario described explains an intervention period activity, namely, patient follow-up, monitoring and exit.

Pre-intervention period: User recruitment, assessment and therapeutic plan

Dr. Ross participates in the m-RESIST project. He logs in the m-RESIST dashboard, resets the password and accesses to his personal profile. He updates the details of his basic profile, and can access to help pages to find more information on how to use the system, FAQs and user support tools.

Fred Hughes, diagnosed with Schizophrenia, has been treated in Dr. Ross' Hospital for a long time, but positive symptoms are persistent. As he has been considered to meet criteria for treatment-resistant schizophrenia, and he is cooperative and used to digital technologies, Dr. Ross offers him to be included in the m-RESIST project.

In the first appointment, Dr Ross accesses the dashboard and makes a starting assessment, uploads the patient's data to m-RESIST, and set a new profile for Fred. He is given the smartwatch and m-RESIST app is being configured on his smartphone. Dr. Ross educates Fred and Mrs. Hughes (Fred's mother) in using the m-RESIST solution. Dr Ross sends a message to Fred to check messages service works. He receives the message and through the app replies to Dr. Ross. Finally, Dr. Ross schedules the next appointment with Fred. He receives a message through the app reporting this new appointment. During the next two weeks, Fred uses his smartphone and smartwatch. He receives a reminder by the app regarding the next visit with Dr Ross.

Before the next visit with Fred, Dr. Ross logs into the dashboard and accesses the list of included patients with a brief overview (new messages; module of treatment activated) ordered by "Level of Risk". Dr. Ross selects inbox and filters messages by priority, seeing messages with read/unread state. Dr. Ross accesses individual messages and deletes unnecessary ones. Dr. Ross selects Fred's profile and has

access to the next information: level of risk, module of treatment activated, messages, chart with ongoing sensor data. Dr. Ross composes and sends message to Fred on a weekly basis.

On the next visit, Dr. Ross logs into the dashboard fills variables in the patient schema. Dr. Ross accesses and edits early warning signs and coping strategies table for Fred. Dr. Ross scores early warning signs, and gets by the system the definitive list of early warning signs and coping strategies for Fred. Dr. Ross sets the list of problems and activates a module of treatment (Symptom Management and Healthy Lifestyle). Dr. Ross accesses, browses and views in detail psychoeducational content. Dr. Ross updates psychoeducational content and Fred can access this content though the app. Dr. Ross sets a new appointment (online) with Fred.

Two days later, Fred feels that the side effects are worsening. He sends a message through the m-RESIST app to Dr. Ross and asks for a new appointment of an online visit. Fred and Dr. Ross make the scheduled online visit. They decide together to set the Treatment Adherence Intervention and stop the Healthy Lifestyle Intervention through the dashboard.

Intervention period: patient follow-up, monitoring and exit

One month after being recruited for m-RESIST, the system (CDSS) “detects” a significant deviation in Low Activity Level (sensor module raw processed data). Fred receives a message and the questionnaire “Need4Help”. Fred marks he needs Dr. Ross help, so a message is sent to him through the dashboard, while the system (CDSS) asks for a set of recommendations (to the Recommender), based on previous feedback and predictive variables (Predictive Module), which are then delivered to the patient by the app. Three days later, Fred receives a message asking for his current state. Fred uses the app to answer and receives a message of reinforcement.

One week later, Fred feels agitated, so he asks for the “Need4Help” scale through the app and choses to be helped by the Dr. Ross. Dr. Ross receives the alert in his dashboard and sends a message to Fred. He wants to set an appointment for a regular, face-to-face visit, so Dr. Ross schedules the appointment in the m-RESIST website (dashboard). Dr. Ross keeps a specific monitoring on Fred’s data in the dashboard, looking at the questionnaires results sent to Fred. During the coming weeks, Fred’s smartwatch data seem to come back to their normal levels. From time to time, Dr. Ross sees some discontinuation in the data, and sends a reminder to Fred through a message, asking whether he still feels comfortable with the use of m-RESIST. Fred responds to the message that sometimes he doesn’t like to wear the watch all the time, although he is quite comfortable with the contact he maintains with Dr. Ross through the app. Dr. Ross then focus his monitoring actions towards Fred more focused on questionnaires than sensor data reflected in the dashboard.

Two months later, James, another patient of Dr. Ross that joined the m-RESIST program at the same time than Fred, experiences some difficulties with the use of the smartphone and smartwatch, and does not feel comfortable anymore with the program. Dr. Ross, after receiving a call from Sarah (James’ caregiver, his sister), checks last values recorded in the dashboard through the sensors, and observes that data collected is very discontinued. Dr. Ross appoints Fred and Sarah, and in a face-to-face meeting they agree that James should leave the program.

3 m-RESIST Components

The purpose of this chapter is to explain briefly functionalities that are implemented by the components, focusing on how it fits to the scenarios described above.

3.1 Dashboard

The dashboard consists of a set of UI components that visualize and edit the data provided by the back-end systems, and allows the users to interact with the back-end systems. Regarding the accessibility, it requires authentication and it's accessible only by the administrator and the clinicians.

The dashboard uses WordPress content management system (which is based on PHP and MySQL), in order to authenticate and manage the users. The front-end functionalities are implemented with JavaScript and the AngularJS 1.x. framework, whereas the responsive and dynamic UI is achieved with the use of CSS and the Bootstrap Framework. The communication between the dashboard and the m-RESIST web services is implemented with AJAX (Asynchronous JavaScript and XML).

The dashboard provides a set of functionalities to the connected clinicians.

- **See patients' overview/details:** The clinician sees an overview or details of his assigned patients and their current status. The patient's data is loaded from different back-end systems. More specifically the patient profile and the sensor statistics are loaded from mIR, whereas the mails and the appointments are loaded from the mailing and the calendar system respectively. It should be mentioned that the patient profile is stored in multiple indexes and types inside mIR. Of course, as explained above, all data is loaded with AJAX.
- **Edit profile:** The clinician edits his profile which is stored in the WordPress' MySQL database.
- **See full list of users:** The clinician sees a full list of colleagues, caregivers and assigned patients. This list is stored in the WordPress' MySQL database and is based in WordPress plugin that allows extending the default user fields. Additionally, the clinician is allowed to create new caregivers and patients, as well as to edit their profiles.
- **Send and receive messages:** This functionality uses the mailing system and allows clinicians to send or receive mails. The communication with the mailing system is established through the ESB and is made with AJAX.
- **Receive notifications:** The clinician receives real-time notifications which are sent by the notification system. Usually, notifications are received when the clinician has a new message in the inbox.
- **Manage calendar:** This functionality uses the calendar system and allows clinicians to create appointments and see them on calendar. The communication with the calendar system is established through the ESB and is made with AJAX.
- **Manage educational content:** This functionality uses the WordPress Content Management API to provide tools for creating and editing educational content for the Clinicians and the Patients.

- **Access recommendation UI:** This functionality allows clinicians to access the recommendation UI (which is embedded as an iframe to the dashboard) and consequently configure the recommendation engine.
- **View help pages:** The clinician views a full list of functionalities supported by the dashboard, and details on how to use them.

Some of the functionalities explained are pictured from Figure 3 to Figure 9.

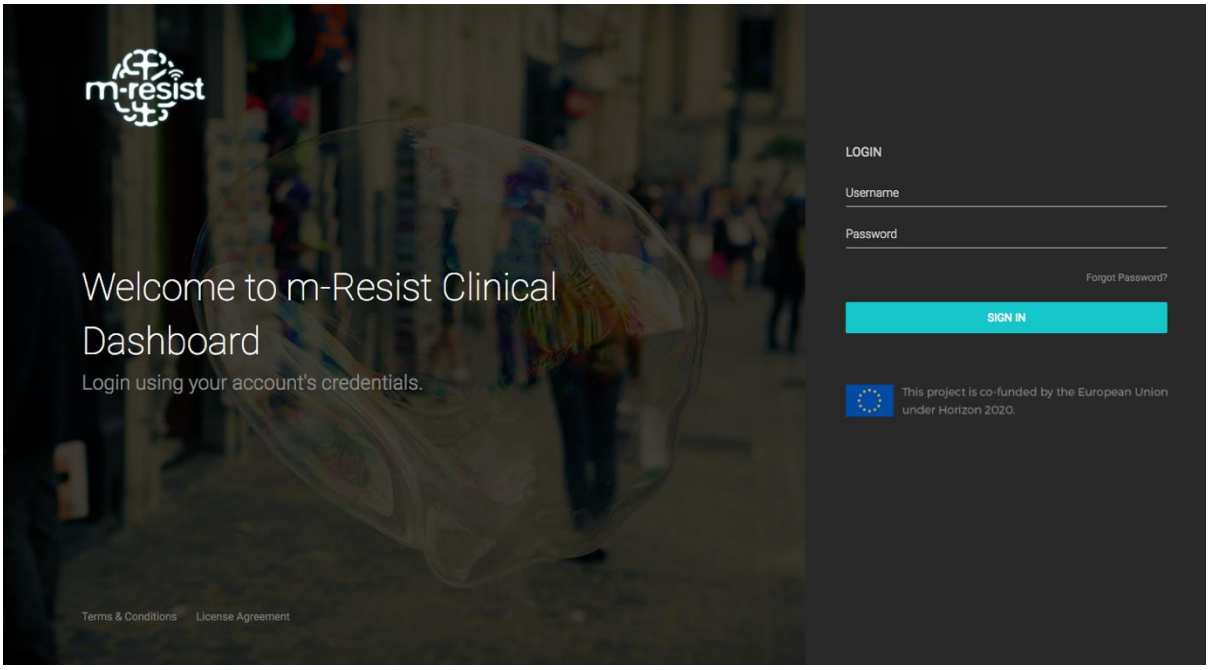

Figure 3 Login page

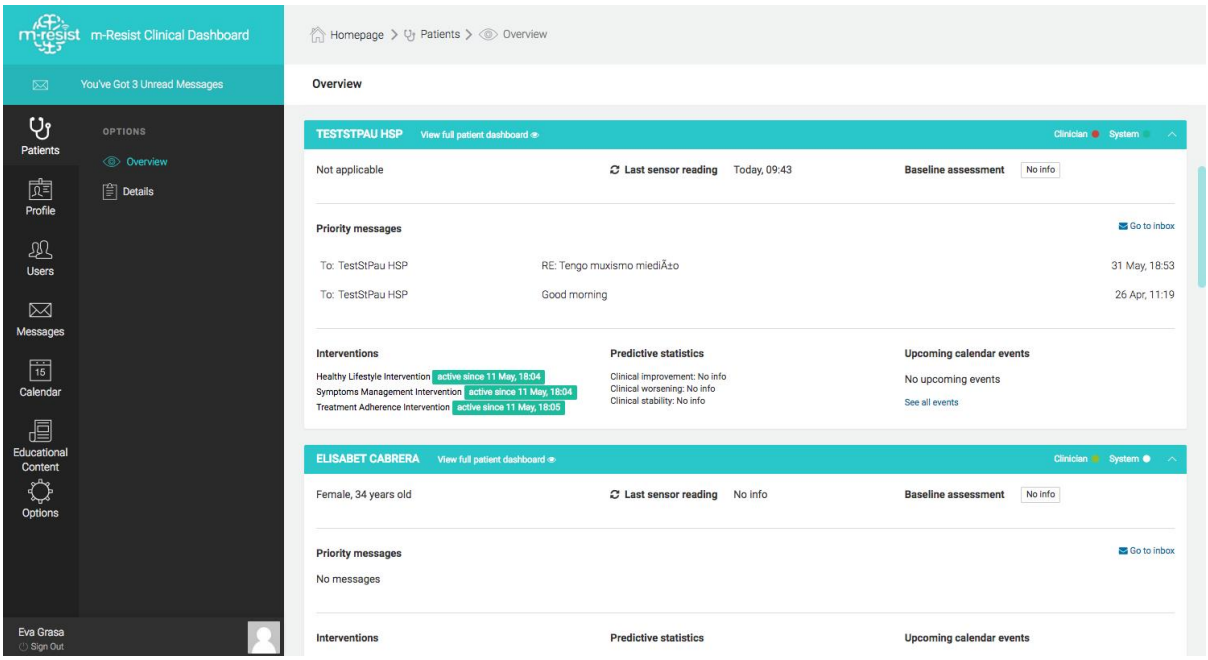

Figure 4 Patients overview

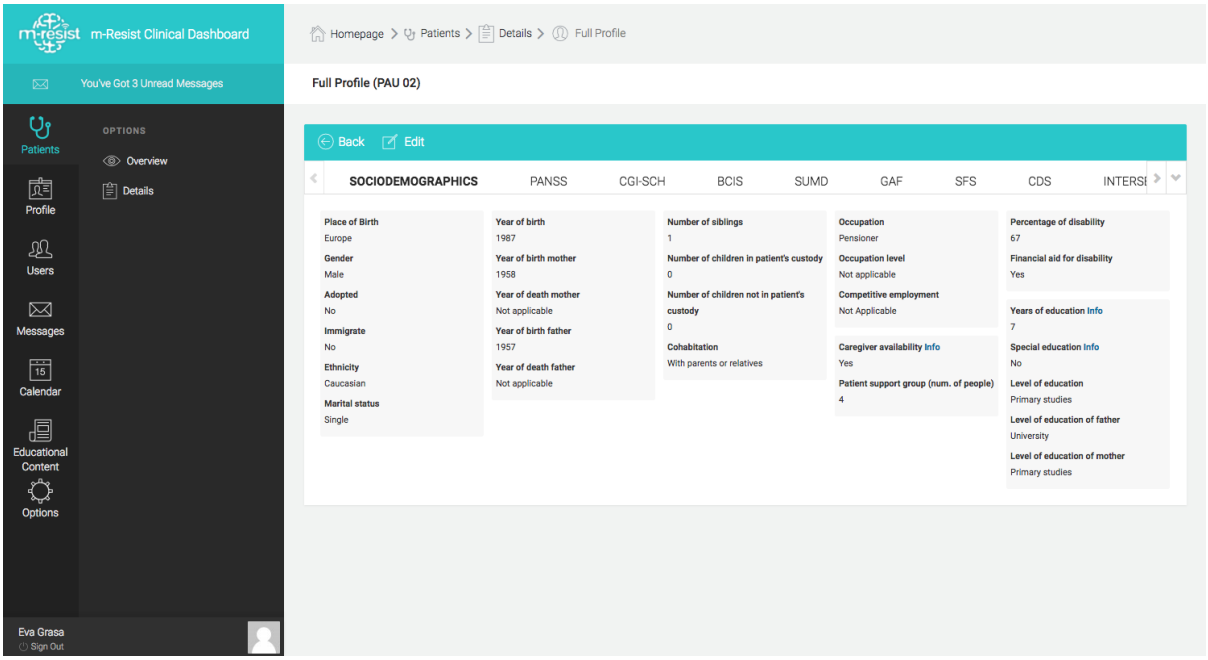

Figure 5 Patient full profile

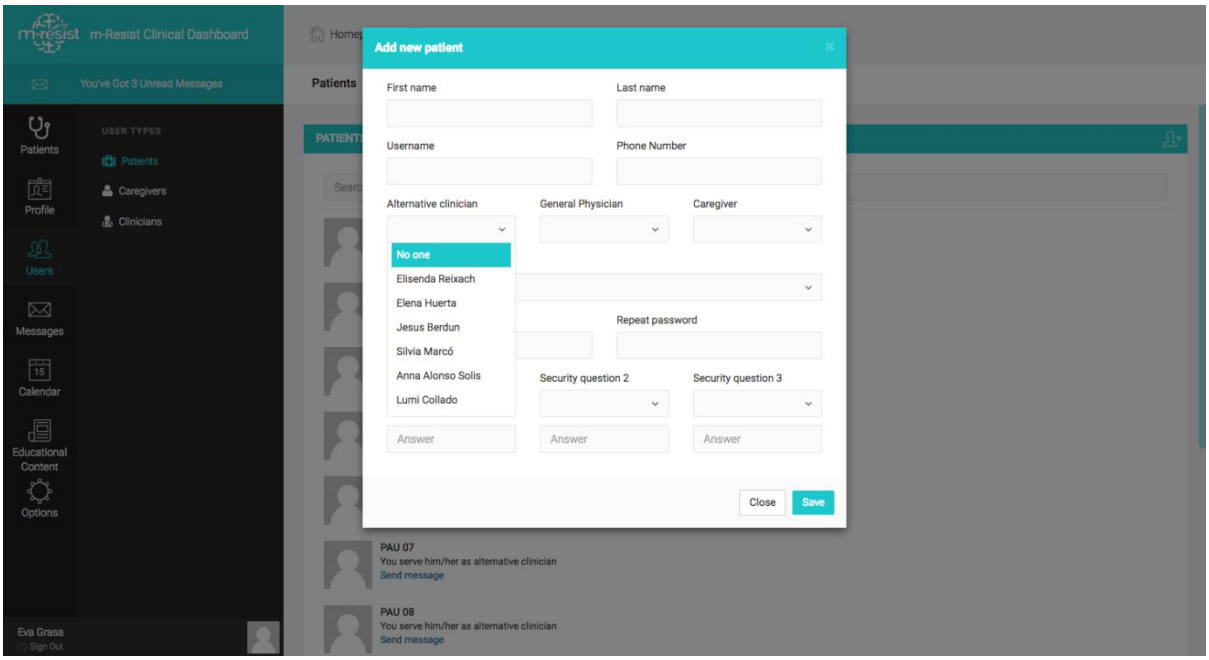

Figure 6 Add new patient

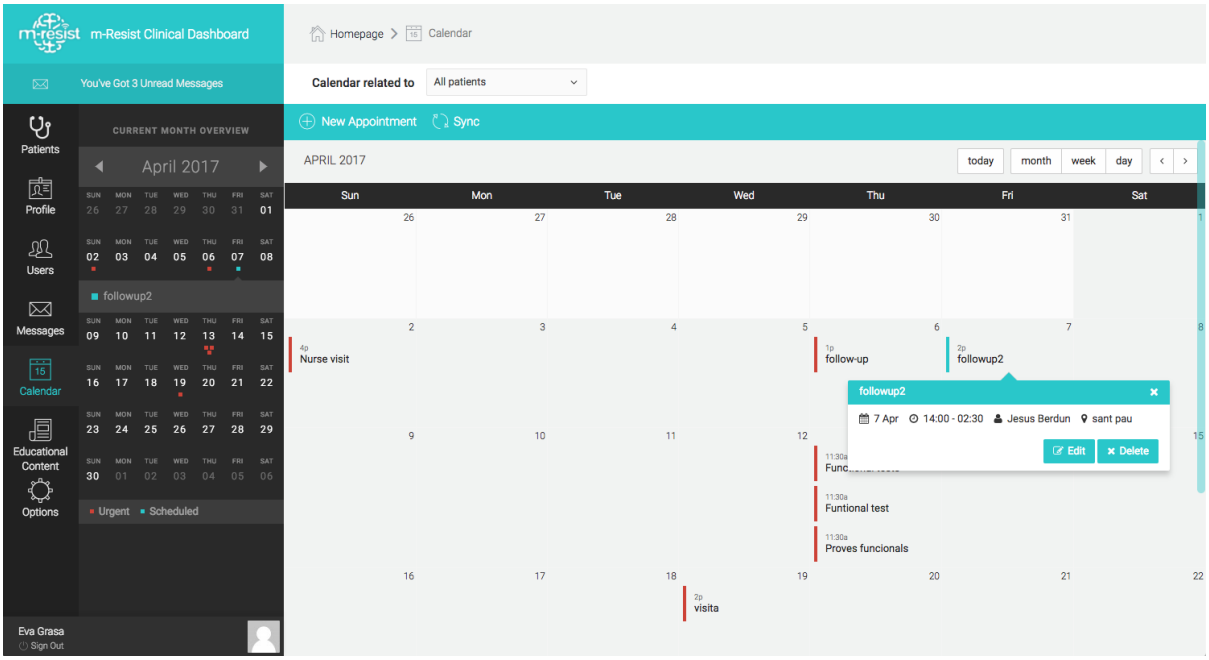

Figure 7 Calendar – Month view

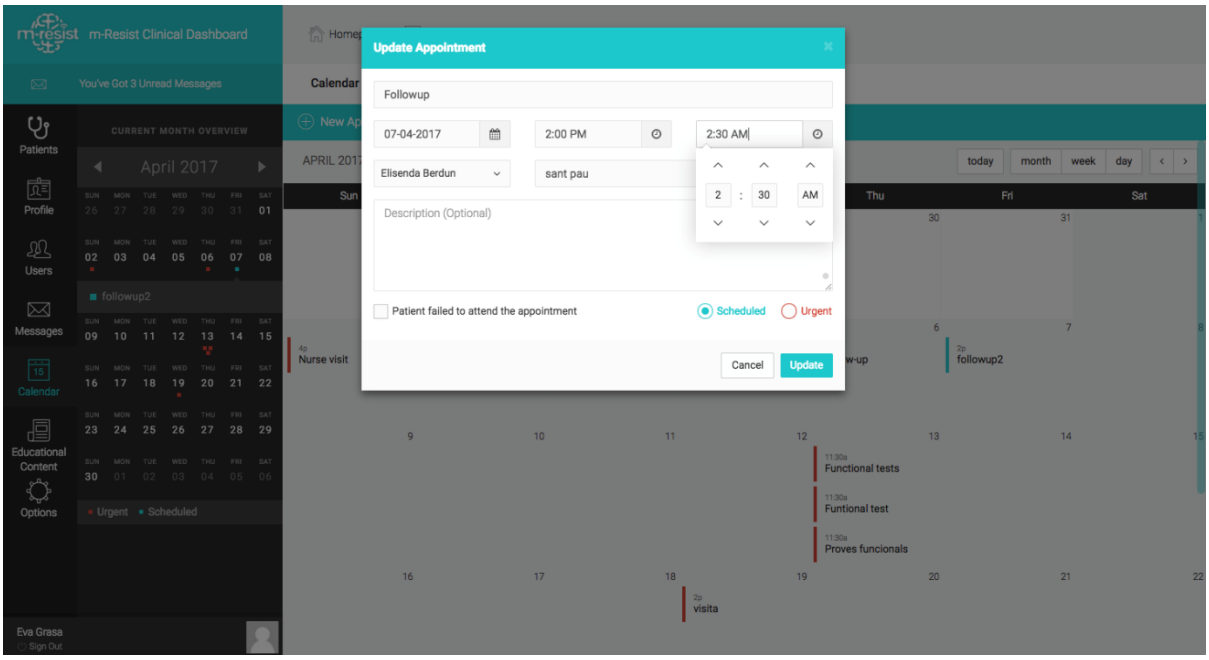

Figure 8 Calendar – Edit appointment

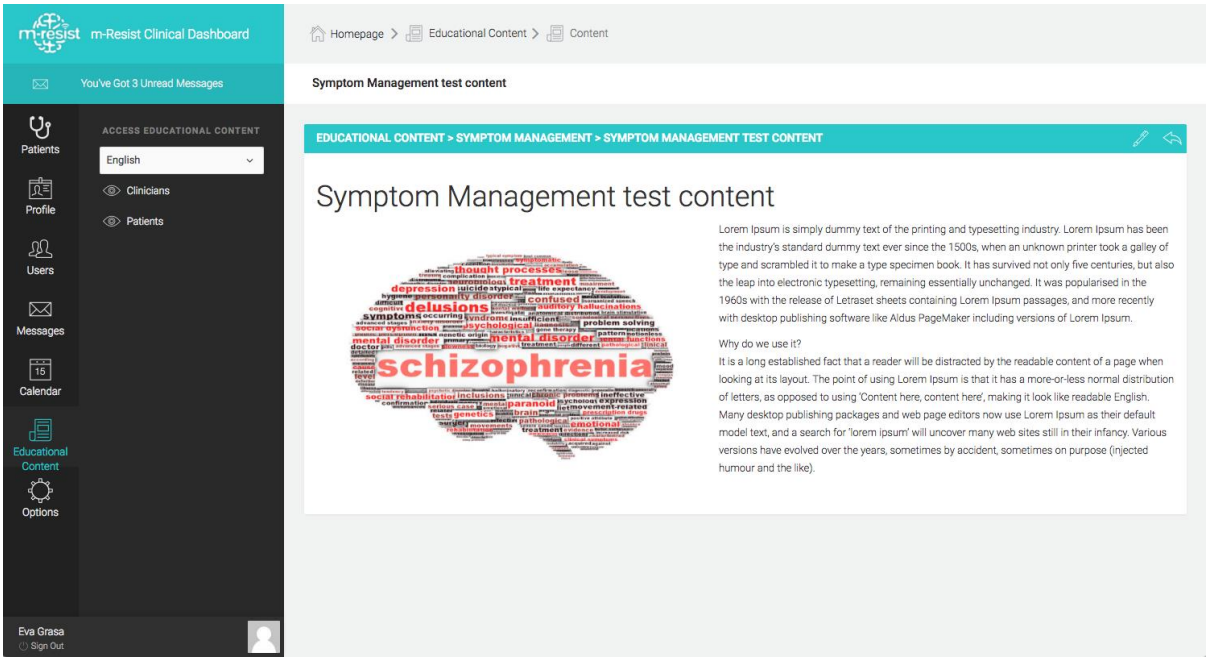

Figure 9 Educational Content – View article

3.2 Mobile Application

The mobile application consists of an Android smartphone app and an Android Wear smartwatch app. The smartphone app is the only interaction medium for the patient with the m-RESIST system.

The patient logs in to the app using the credentials assigned to him by the clinician during his first visit when he was also handed the smartphone and the smartwatch. It is not possible to create an account on the app by itself. The app is currently available for Android 6.0+ smartphones and access is permitted only by invitation. It is available on the following URL:

https://play.google.com/store/apps/details?id=be.iminds.mresist.

The apps are written in native Java code using Android studio. It communicates with the ESB through a REST API. The app provides a set of functionalities to the patients.

- **Authenticate and recover password:** The patient can authenticate himself in the app using the credentials provided by the clinician during the visit where he was added to the m-RESIST system. In case he forgot his password, he can reset it on the login screen using his username and by answering 3 security questions that were set by the clinician together with the patient.
- **Read and send messages:** The patient can receive and send messages using the m-RESIST mailing system. He can reply to messages send by his doctor, case manager or caregiver but not to messages sent by the system. He can create messages and send them to one recipient. The messages are sent and retrieved using the mailing API.
- **See appointments:** The patient can consult his appointments with his doctor or case manager. He cannot create appointments
- **Receive and answer treatment requests:** The patient receives questionnaires from the CDSS based on the interventions assigned to him. The app shows the questionnaires as a multiple-choice question and provides a way to answer them. Questionnaires can consist of 1 or more questions and the patient needs to answer all of them before he can submit the answers.
- **See educational content:** Patients can consult educational content provided by the clinicians through the dashboard.
- **Consult emergency contacts:** The patient can press an emergency button (pictured as a bell) on every screen guiding him to some emergency contact numbers specific to his clinic.
- **Gather sensor data:** The wear app (heart rate, steps) and the smartphone app (GPS, communications, battery) gathers sensor data about the patient and uploads this to mIR to be consumed by other components.
- **Multilanguage:** The app is translated in Catalan, English, Hebrew, Hungarian and Spanish. The selected language is based on the system language. In Hebrew, the app supports a complete right-to-left layout (icons, text, layout).

Some of the functionalities and screens explained above are represented in the following figures (from Figure 10 to Figure 13).

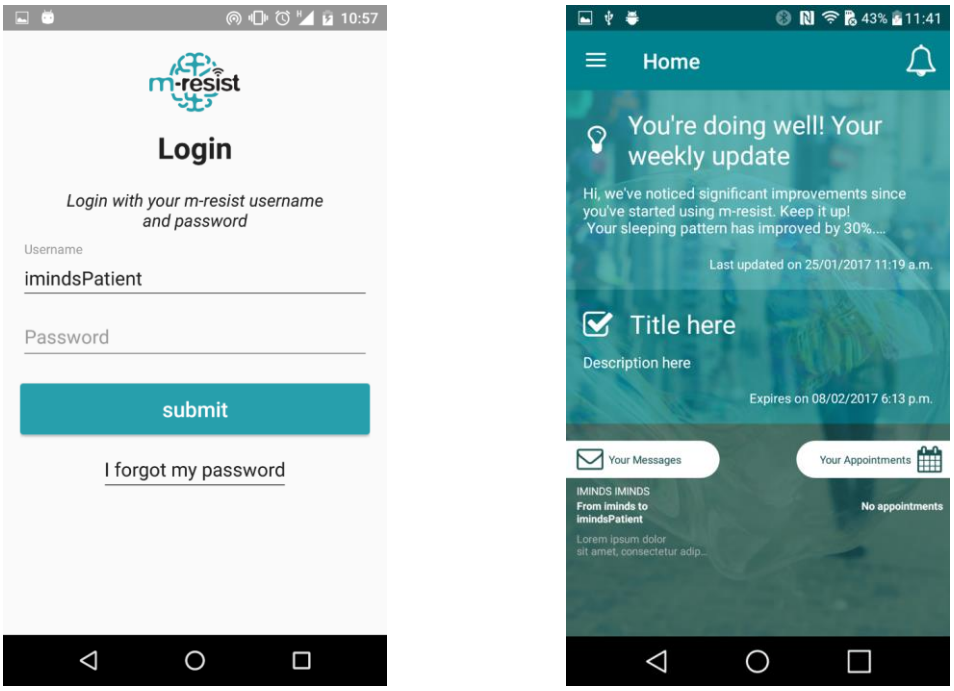

Figure 10 App log-in screen and home screen

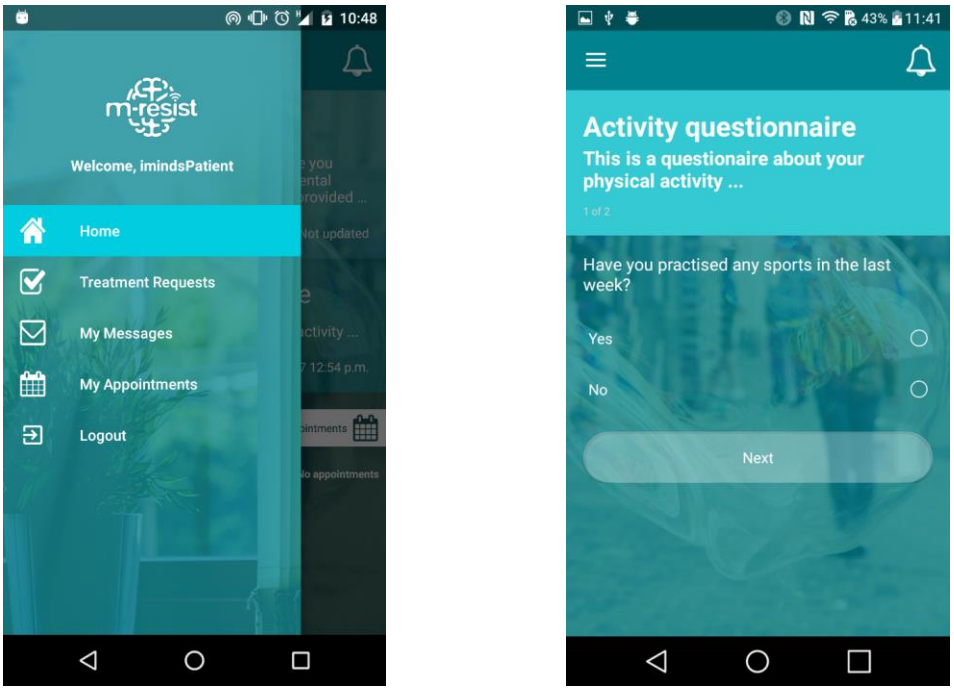

Figure 11 App Menu and Treatment request question

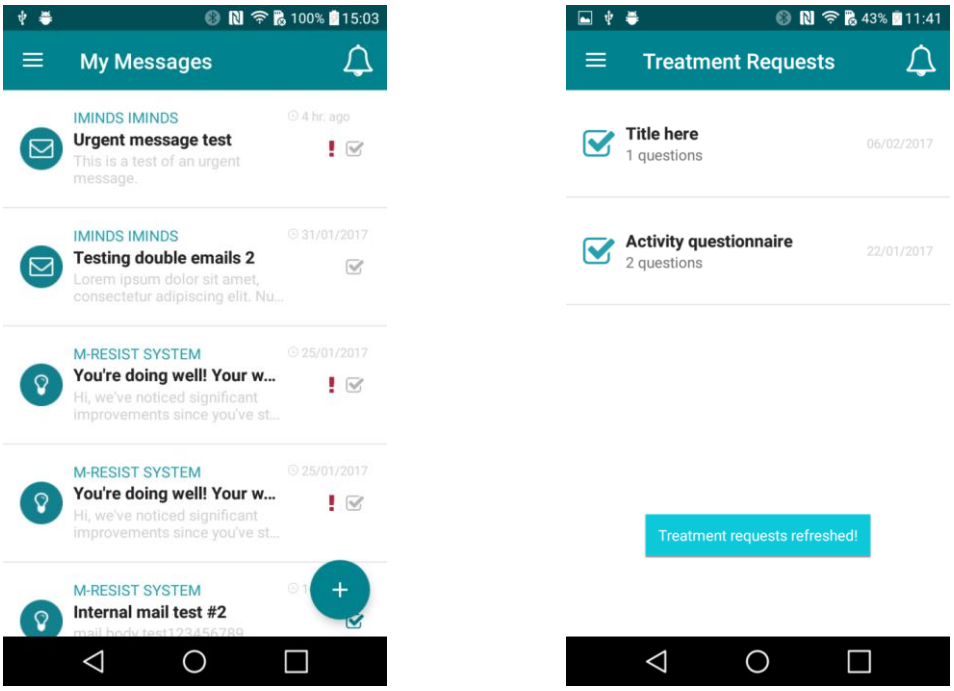

Figure 12. Messages list and Treatment requests

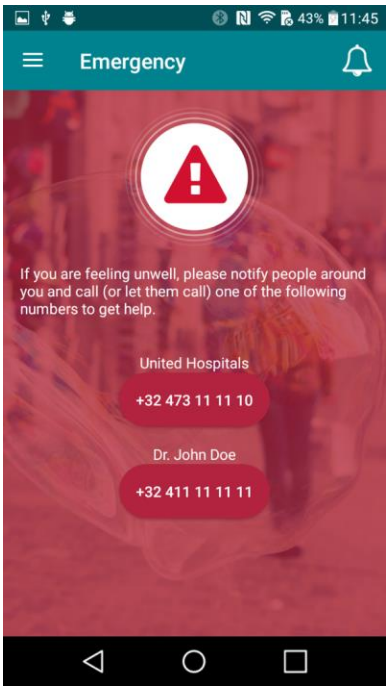

Figure 13 Emergency view

3.2.1 Sensor data

The purpose of the Android Wear app (and partly of the phone app) is to gather sensor data and upload this to mIR for use by other components. To do this, the watch needs to be within Bluetooth connection of the phone at all times (+-10m). When the phone app is started, a message is sent to the watch to check if the watch app's sensors are on. If not, it starts listening for the heart rate and step count. A notification is shown on the watch screen to show that m-RESIST is capturing the sensor data (see Figure 14). On the watch, the sensor data is buffered until it reaches a maximum size (100kb) or reaches a certain age (1 minute). It is then sent to the phone through the Bluetooth connection. All the sensor data is cached on the phone's local storage until it is uploaded to mIR at a fixed interval (every 15 minutes).

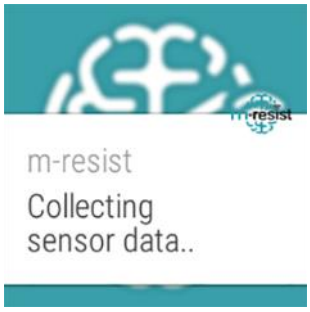

Figure 14 Watch app

3.3 m-RESIST Information Repository (mIR)

The m-RESIST Information Repository (mIR) acts as the central data repository for the entire system. mIR allows external modules to store and retrieve any kind of information and does not impose a particular schema. mIR exposes an HTTP Restful interface, which accepts input and provides responses in the JSON[1] format. HTTP verbs such as PUT / GET / POST / DELETE are mapped to the corresponding data storage actions Create / Retrieve / Update / Delete (CRUD).

mIR is schema less, which means that modules can store any information as long as they are in the JSON format. Each document stored in mIR is of a certain *type*, which is stored in a specific *index.* The *type* and *index* are the basis of the URLs which are used to perform actions in mIR. mIR's API is documented here **http://docs.mirmresist.apiary.io/**

Furthermore, mIR provides module specific APIs, and enhanced query functionality such as search and aggregation algorithms based on the underlying *index* through the use of document conventions. Document conventions ensure that the mIR document contains data in order to adhere to a specific convention.

In essence, conventions act as an agreement between mIR and modules that:

- There will be specific keys in the JSON document (key set)
- There will be specific APIs which will provide extended logic for each convention (convention data).

Depending on the convention, specific documents are validated during saving and updating. During document retrieval, filtering is performed on the conventions, and additional logic is provided based on the selected conventions.

Our implementation allows for conventions to be added at any time, however pre-existing conventions cannot be modified, in order to handle this, convention names need to be updated, or a version namespace needs to be introduced.

A document adheres to a convention if it contains the reserved key "_conventions". This is an array of convention names, where each name provides additional storage validation and query logic.

```
{
 _conventions: [ convention_name ]
 ... document data ...
 (convention keyset: convention data,)+
}
```

Figure 15 Snippet Document Conventions

Each convention may require a set of document *keys* to be available in the document adjacent to the rest of the data, these are called the *convention keyset*. Furthermore, through the use of conventions mIR provides specific endpoints which leverages these conventions. Finally, a convention can also affect the response payload of a PUT or POST request.

3.4 ESB / Message Bus

The integration layer of m-RESIST back-end is consisted of 2 core components: the WSO2 ESB and the WSO2 Message Bus. All the functionality that is exposed by the various back-end components, the mobile applications as well as the dashboard passes through the ESB promoting thus an event driven architecture. By adopting this kind of architecture design pattern we make sure that the resulting system is flexible enough to support the loose coupling of the various components.

3.4.1 ESB layer

The ESB layer is composed by a number of proxy services that encapsulate the actions that have to be performed transparently each time the m-RESIST service APIs are accessed. A proxy service is a virtual service that receives messages and applies various transformations and other functions before dispatching them to the target API endpoint. The building blocks of an ESB proxy service are called mediators. Custom ESB mediators have been developed and integrated to satisfy the needs of the back-end components. Some of the requirements that have been addressed are the follows:

3.4.2 Mail component

The API of Mail component has been extended from its initial implementation to support additional functionality. Some of the new functionalities include:

All API methods exposed by the Mail component require a valid JWT token in the headers in order to authenticate the requesting subject. Moreover, in some situations the role of the requesting subject is also extracted from the JWT token to also support a role-based authentication (clinicians, patients, system etc) in order to restrict access on the exposed information.

The API has been enhanced by implementing new methods based on the needs of the integrated system. Some of the functionality that was added includes:

- The ability to search by filtering on some desired tags (that are part of a mail). Each mail now can be characterised by a tag, providing components the ability to search mails based on it. As such, communication exchanged with users for a specific action i.e. Healthy Lifestyle Intervention can be identified and exploited in the means of m-RESIST.
- The ability to get a whole conversation by supplying any mail id that belongs to that conversation. The results are sorted by date, starting from the original mail and then all replies are following.

3.4.3 Calendar API

The calendar system exposes a RESTful HTTP API that is comprised of CRUD operations on the appointment resource as well as a search API to effectively retrieve calendar appointments by date. All API methods exposed perform a JWT token validation permitting access to only authenticated and authorized users.

The backend system of the mail controller has been implemented using the Oracle JDK 8. The web layer was extended using the Spring web project. It is independently deployable with respect to the rest modules of the system promoting thus a microservices approach. For the deployment of the web services exposed, a Tomcat 8 web server and container is used.

The calendar system offers web services to allow other external factors like the m-RESIST components to interact with its core engine. These services are exposed from the calendar system to the ESB and the ESB undertakes the responsibility to publish them to the message broker.

Having the ESB interfere with the calendar system is not a coincidence. In the means of m-RESIST, all communications will pass through the ESB mainly for two reasons:

- ESB supports mediation; any authorized component is allowed to communicate with the calendar system with respect to the API supported by the calendar system. In case a component cannot collaborate with the API provided by the calendar system, ESB can mediate and convert a set of data values from the data format supported by the calendar system into the data format supported by this component.
- Communication logging; any communication request can be logged in order to implement an auditing mechanism. Due to the nature of the m-RESIST project, the communication monitoring is considered a crucial success factor. To that end, ESB can log all communications in order to

provide useful information to system administrators, verify the security and integrity of the m-RESIST platform as a whole and identify potential security holes or vulnerabilities.

3.4.4 Notification API

The notification system provides functionalities that allow both users and components to send instant push notifications to subscribed end users. Technically, it sends push notifications via the following possible ways:

- As internal mail using the provided Mailing System
- As push notification using the Firebase Cloud Messaging

The end user has to subscribe to the system in order to receive a notification. The subscription is feasible from both web and android platforms, and the subscribed user could see a popup for every received notification.

The system is accessible through a RESTful API that exposes services for all the above functionalities. As for security, all API exposed methods, perform a JWT token validation permitting access to only authenticated and authorized users. The notification system has been implemented using the Oracle JDK 8. The web layer was extended using the Spring web project. It is independently deployable with respect to the rest modules of the system promoting thus a microservices approach. For the deployment of the web services exposed a Tomcat 8 web server and container is used.

3.5 Sensor Module

The Sensor Module performs the digital processing over the raw data time series recorded by the App and computes the processed values needed by the CDSS and the dashboard (listed in D3.4) for visualization purposes. ESB messaging allows the synchronization between Sensor Module, App, and CDSS. Table 1 shows sensor data collected by the m-RESIST system.

General Parameter	Extracted Parameters	Description	Data Structure	Data/Source	End user
Time Spent Outside	Time Spent Outside	Overall time spent outside during the 24h	Single numeric value	GPS	CDSS
	Location	Shows where the person was during the 24h (0: outside, 1: at home).	Array of values [0 or 1]	GPS	Dashboard
	Distance	Shows the distance travelled in time during the 24h.	Array of numeric values	GPS	Dashboard
Global Activity Time	Low Activity Time	Overall time spent in low activity during the 24h	Single numeric value	Accelerometer (Gyroscope) *Usage subject*	CDSS

				to usability	
	Activity Class	Shows the activity class performed in each moment during the 24h (0: low activity, 1: non-low activity).	Array of values [0 or 1]	Accelerometer *Usage subject to usability*	Dashboard
Global Activity Index	Activity Amplitude	Parameter that gives the amplitude of the accelerometer in time	Array of numeric values	Accelerometer *Usage subject to usability*	Dashboard
Deep Sleep Time	Time Non Rem Sleep	Overall time spent in deep sleep during the night.	Single numeric value	HR (Accelerometer)	CDSS
	Sleep	Shows the sleep phase in time during the sleep (0: deep sleep, 1: rem, 2: awake).	Array of values [0,1,2]	HR (Accelerometer)	Dashboard
Light Sleep Time	Time Rem Sleep	Overall time spent in rem sleep during the night.	Single numeric value	HR (Accelerometer)	CDSS
	Sleep	See above	See above	HR (Accelerometer)	Dashboard

Table 1 m-RESIST sensor data

Following there is a list of sensor module functionalities

- Receive ESB messages from the topic ***mresist.sensor.raw.all*** that are generated every time raw data from accelerometer, heart rate, and GPS acquired in a certain time interval (with normal connection, around 15 minutes) are stored in mIR.
- Retrieve the corresponding **patientId** and **endTime** (unix milliseconds of the last record) from the "_link" in the message received
- Retrieve from mIR home coordinates (latitude and longitude) input in the dashboard in patient registration phase.
- Check whether it is the first time for that specific **patientId** and, if that is the case, generate a record for it. If the **patientId** is within the patientIds already processed the software retrieves from mIR and stores in local variables the last values saved for: **lowActivityTime, remTime, nonRemTime, timeSpentOutside**, along with the **startTime** and the parameters obtained during the training phase (mean heart rate and home coordinates). The software needs to retrieve previous values of **lowActivityTime, remTime, nonRemTime,** and **timeSpentOutside** because the CDSS needs them as cumulative values.
- Perform a date range query by patientId in order to retrieve the raw sensor data for accelerometer, heart rate, and GPS.
- Parse the returned JSON for each sensor and store the values in specific data structures (see D3.5). Due to some repeating values in raw sensor data from the App, the Sensor Module performs a control over the timestamp of each parsed time series, in order to reduce computational time and occupied memory.

- Process the raw sensor values using the algorithms reported in D3.5 and compute: **timeSpentOutside** (seconds spent outside home for that time range), **timeNonRem** (seconds of non rem sleep for that time range), **timeRem** (seconds of rem sleep for that time range), **timeLowActivity** (seconds spent in low activity for that time range), **location** (array with 0: outside, 1:at home), **distance** (array of distances in meters), **activityAmplitude** (array of accelerometer amplitude values), **sleep** (array with 0:non rem sleep, 1: rem sleep, 2: awake).

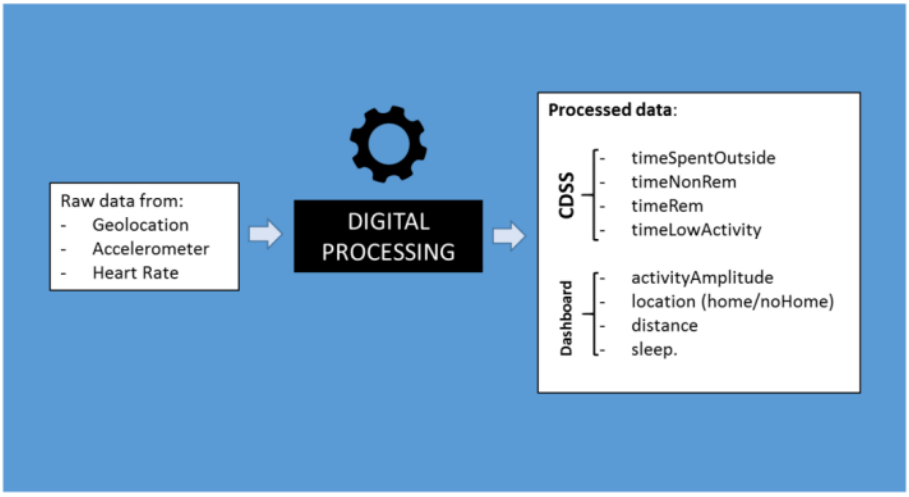

Figure 16: Sensor Module

- Update the values for **lowActivityTime, remTime, nonRemTime, timeSpentOutside,** adding them to the previous values (the CDSS uses them as cumulative values).

- Put the cumulative value for **lowActivityTime, remTime, nonRemTime, timeSpentOutside,** publishing to the topic ***mresist.sensor.processed.***

- Put in mIR the time series for **location, distance, activityAmplitude, sleep**

The Sensor Module needs some training parameters in order to perform the processing over the raw sensor data. These training parameters are **home coordinates** and **mean RR interval** (and consequently mean HR). Home coordinates are input by the clinician from the dashboard after the patient consent, whereas the mean RR interval is computed dynamically by the sensor module: the initial value is 0.85 (population average) but this default value is updated every 200000 RR samples (around 5 days with 0.5 Hz sampling) with the average value computed from the user's data.

Along with the other patient info, the system stores the cumulative value of the RR for the specific user and the number of samples the cumulative value corresponds to. After a certain number of samples that

corresponds to about 5 days of usage with no gaps in RR time series, the sensor module computes the mean of the RR interval dividing the cumulative value by the number of samples and makes an average between the value obtained and the old value stored. In this way the mean RR value is not computed once in the beginning, but keeps updating slowly (weekly based) following the possible user fluctuation of RR values and being, in this way, more user specific.

The Sensor Module works as an autonomous Java application that has been deployed on the m-RESIST windows server given by ATC. In order to evaluate the correctness and robustness of the processing algorithms and the overall processing flow there is the need of continuous testing of the completeness and integrity of the raw data generated by the smartwatch and smartphone, completeness of time series returned by the mIR queries along with the refinement of processing algorithms parameters.

3.6 Predictive Modeling

The predictive module provides predictions for different measures of a patient's status based on the information currently available from the patient. Although there is no data available from the m-RESIST project yet, the module contains a set of predictive models that have been pre-trained with patient data provided by the clinical partners (a total of 450 patients).

The variables available as input for the models are the following:

- Gender	- Level of education	- Marital status
- Living environment	- Employment status	- Number of relapses
- PANSS (P, N, G)		

The models receive these variables as input for each patient and output probabilities for each of the different possible outcomes of a list of target variables defined by the clinicians. A set of models have been developed and deployed for the following target variables:

Target variables	Possible values	Description
Evolution	Improvement / Stable / Worsening	Describes the change in PANSST scale of a patient with respect to basal.
Low insight	Yes / No	Describes if the patient presents low insight.
Resistance	Yes / No	Describes if the patient is resistant.

Table 2: Target variables

Based on the available input variables, the models output the patient's probability of improvement, worsening, remaining stable, presenting resistance and presenting low insight in the following months. These values are made available to the rest of the modules by storing them in mIR. When any of the input variables of a patient is changed, these probabilities are computed again and posted to mIR through the ESB.

3.7 Clinical Decision Support System (CDSS)

The CDSS is designed to provide the users with necessary information to support health-related and clinical decision-making. The system utilizes available data sources, in order to assess the patient's condition using decision algorithms, and, as a result, classify the clinical conditions in order to allow providing clinical and lifestyle recommendations available in the Recommender component.

The CDSS Module analyses the various passive and actively collected inputs and generates rule based interventions to the patient. The CDSS starts with a training period of two weeks, to accumulate the patient baseline and once trained, it monitors the changes vs the baseline. The CDSS is in communication with all modules using various communication technologies to deliver notifications and questionnaires to the patient, the caregiver and the clinical team.

The functionality of the CDSS is based on the workflows developed by the clinical team, reflecting the process of interaction between the system and its users, in order to establish new healthcare pathways. The CDSS activation is triggered by an **event** (i.e., change in the baseline value), that is interpreted in a context of additional information that exists regarding a specific patient, and a series of pre-defined conditions and actions.

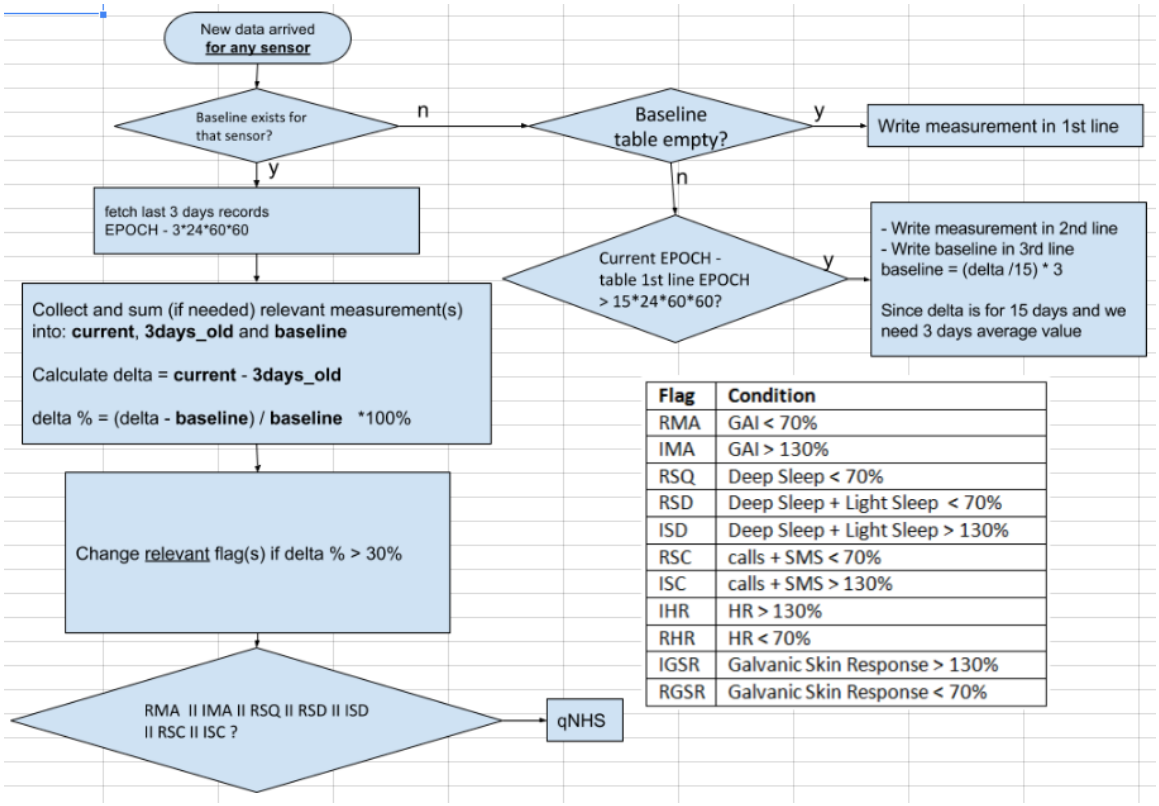

Figure 17 CDSS process

The general starting point of the CDSS activation is a deviation of 30% in one or more of the sensor parameters for the period of 3 days. The process of receiving new sensor data is described using the diagram in Figure 17. The aforementioned change will trigger assessment using a Need for Help Scale (NHS) to be completed by the patient through the m-RESIST app. The aim of this scale is to promote empowerment and self-management. In front of a situation of possible risk, and before to alert clinicians and/or caregivers, the system offers to patient the chance of ask for help, and decide which kind of help needs. Each patient's choice on the NHS activates one of the following **Basal flows** (series of actions reflecting the system's operation in one of the following situation, reflecting the patient's feeling of needing help at the particular moment):

Default risk level	Item
1	Can cope
1	Need mobile help
2	Need GC support
3	Need professional help
4	Need urgent help – Emergency!

The Need4 help scale is connected with the Risk Level scale, so depending on the answer given to the scale, the level could change. All patients included in m-RESIST are allocated into a level of risk that is useful to classify the patient's general condition. The classification consists of 5 risk-levels.

Score	Risk level	Example situations	Frequency of assessment	Modules
0	Baseline	No known risk. Normal functioning	Normal (defined by system)	Baseline modules
1	Low risk	Deviation detected, but patient says he can cope by himself	At least once every 15 days	Baseline modules
2	Medium risk	Patient can cope with mobile help	At least once every 7 days	Risk modules
3	High risk	Needs professional assistance	Once every 3 days	Risk modules
4	Emergency	Life threat or safety risk	Inmediate response	Risk modules

All patients that start m-RESIST for the first time are classified with a risk level "0" or basal, and the basal modules of intervention are activated. In case of possible risk situation, first the situation is assessed by questionnaires sent to patients and, depending on the inputs, the risk level of the patient could increase or maintain.

In case of choosing option "Need mobile help/ Need caregiver support", the m-RESIST app sends a scale to assess risk behaviors. This is called Risk Scale and is based on some items of the Positive and Negative Syndrome Scale (PANSS) and some items of the Calgary Depression Scale for Schizophrenia (CDS for Schizophrenia). The scale has three main thresholds depending on the scores. Whenever one of these conditions is met, the Risk Scale will activate the Symptoms Management Module-Risk and pause the launched basal modules. If the Risk Scale score is different, then the basal modules remain without changes. The CDSS output can be divided into:

1. **Questionnaires** – a series of questions requiring user's response, from which results are calculated, interpreted and communicated to other system modules, such as the Recommender. Each questionnaire has been translated to Spanish, Catalan, Hungarian and Hebrew.
2. **Notifications** – messages that are sent to the user according to interpretation of the system input (i.e., sensor data) and the user's response to the questionnaire (i. e., reply or expiry). Each notification has been translated to Spanish, Catalan, Hungarian and Hebrew.

3.8 Health Recommendation Module

In order to provide recommendations the system follows this sequence

a) How to provide recommendations
b) When to provide recommendations
c) **What to provide as recommendations**

The Health Recommendation Module (recommender) is responsible for **what** to provide.

Due to privacy restrictions concerning historical data we could not create recommendations implicitly, therefore our implementation provides:

- An interface which allows clinical experts to populate the system with recommendations items.
- The recommendation engine which contains the necessary algorithms to provide recommendations
- A feedback mechanism so that recommendations can be personalized based on the patient recommendation selection.

The recommender provides a user interface which allows the clinical experts provide and classify recommendation through a dynamic set of classification criteria. This information is modelled as a 2-dimmensional vector containing the criteria used to classify a recommendation item, as well as its values (see Figure 18).

Name	Criteria 1	Criteria 2	Criteria 3	Criteria 4
Type	**range**	**boolean**	**range**	**boolean**
Values	3 - 6	0	10 - 100	1

You should take a walk outside

Figure 18 Recommendation Example

Recommendations can then be queried using a 2-dimensional vector containing the criteria and its explicit query values, as shown in Figure 19.

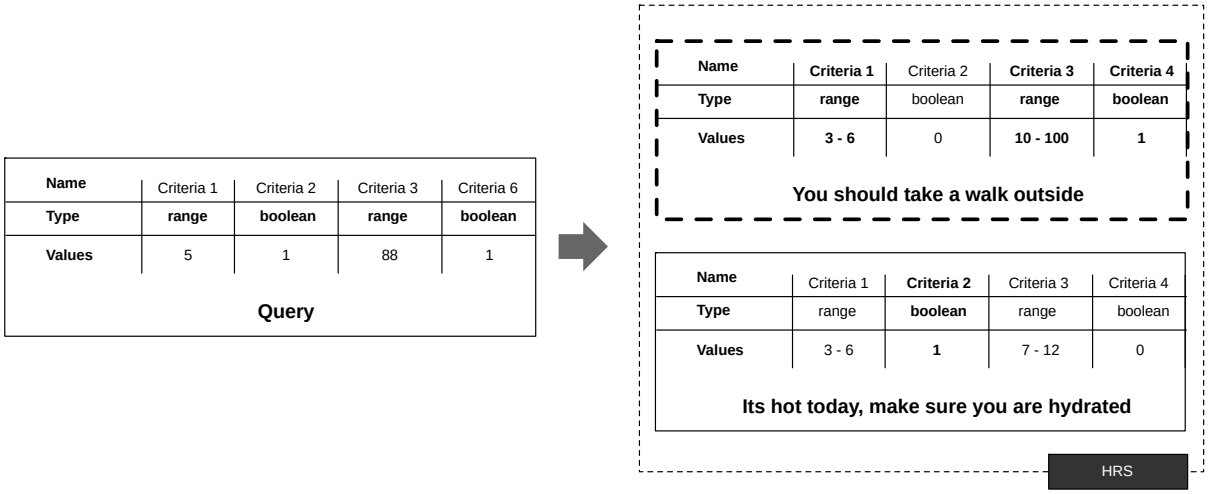

Figure 19 Recommendation Query

In the scenario in Figure 19, both recommendations would be selected due to the fact that they both have matching criteria found in the query, however *"You should take a walk outside"* would rank first in the recommendations returned as a response.

Personalized Recommendations

However due to the fact that multiple recommendation items can be grouped under the same classification criteria we have to introduce a mechanism by which these recommendations items can be ranked and selected differently depending of the patient for which the recommendations are being queried for.

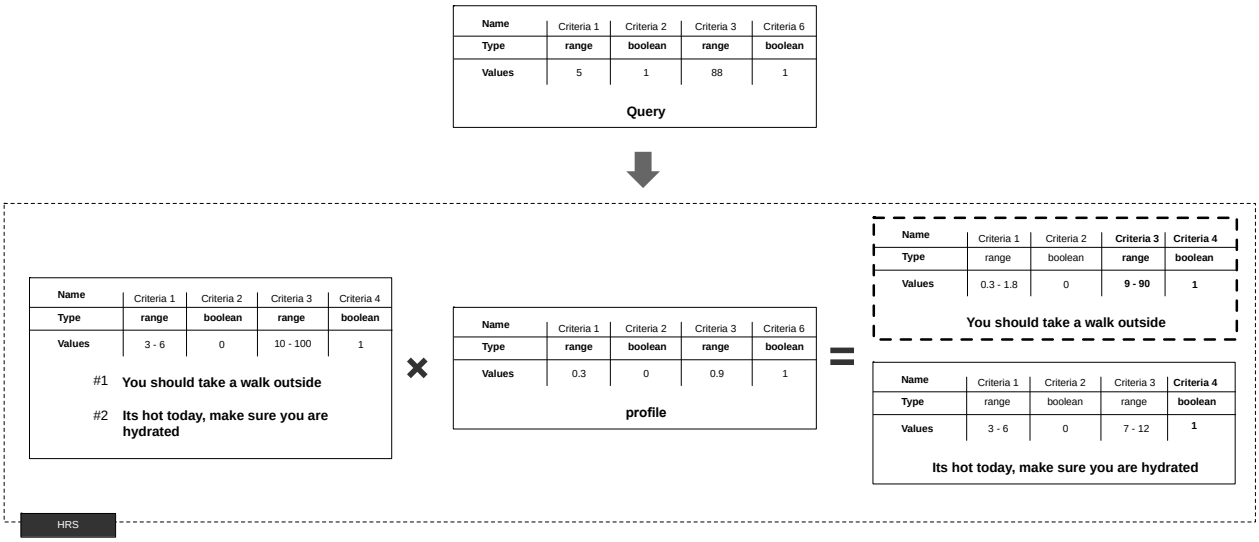

Figure 20 Recommender Personalization

The recommendation algorithm works as follows:

Consider the recommendation item ***d*** containing criteria c_i with values j_i, defined as $d(c_i,j_i)$, where j_i is the range $\mathbf{v_i}$ with minimum value ***m*** and maximum value ***n*** is defined as $v_i(n,m)$ where $n < m$.

The query ***q*** with criteria c_i and value j_i is defined as $q(c_i,j_i)$.

In order to compute a rank we need to the determine the weight ***w*** that query criteria $q(c_i)$ has on each recommendation item $d(c_i,j_i)$. We do this by computing the difference between the query value $q(j_i)$ and the minimum $v_i(n)$ and maximum $v_i(m)$ range value of $d(c_i,j_i)$ so that query values closer to the upper range of the of the classification criteria value $v_i(n,m)$ carry more weight. As a result the weight ***w*** for the query ***q*** with criteria c_i and value j_i is computed as follows:

$$w(c_i,j_i) = (v_i(m) - v_i(n)) / (v_i(m) - q(j_i))$$

The patient weight ***pw*** for each query criteria is $pw(c_i,j_i)$.

We can then use the resulting weight $w(c_i,j_i)$ to compute the ranking of the recommendations. The ranking ***r*** is the sum of all the weights $w(c_i,v_i)$ multiplied by the patient weight $pw(c_i,j_i)$.

$$r = sum(w(c_i,j_i) \times pw(c_i,j_i)$$

where 0 <= ***i*** < available number of criteria in the query.

3.8.1 Recommendation UI

Based on the functionality of the recommendation engine we created a UI which allows the clinical experts to:

- Manage the classification criteria which will be used to classify recommendations, and consequently used by the CDSS to perform recommendation queries. There are two types provided, range criteria, and boolean criteria.
- Manage recommendation categories which are used only throughout the interface and allows the clinical experts to group and organize recommendations.
- Manage recommendation item groups

  Clinical experts are able to add, remove and update recommendation items. They use classification criteria to classify recommendations, by setting specific values to the range and boolean criteria

- Provide language specific recommendations.

The recommendation UI is available through the Dashboard Component, however we provide an open demo at:

http://mresist.imuresearch.eu/ui/

4 Conclusions

A holistic overview of "*m-RESIST Prototype V0*" is provided in this document and shows an improved version of the first m-Resist Beta Prototype after testing activities performed during a pre-trial phase with healthy users in WP4. m-RESIST is a mobile ICT solution addressed to empower patients with treatment-resistant schizophrenia (TRS), that encourages the patients and caregivers to actively participate in the therapeutic process and enables them to self-manage their condition. The document describes an overview of the system followed by use case scenarios explaining how different components of m-RESIST come into play. It is also described in detail the functionality of each component keeping close reference to the scenarios in order to clearly represent the work achieved and the state of the m-RESIST Prototype V0. This version of the m-RESIST system will be used for the piloting activities with real patients in WP5 in the three piloting countries (Spain, Hungary and Israel).

The future work will be focused in refining and extending, where necessary, the integration of the system as well as the functionality of each individual component leading to a new version and final version of the system "*m-RESIST Prototype V1*". Additional refining of the system will be based on the outcomes gathered during the upcoming user pilots.

Our vision for the "*m-RESIST Prototype V1*" is to have a stable refined prototype, which could be deployed and consumed by institutions outside the consortium, so as to further extend and refine its functionality.